Metabolomic Data Analysis with MetaboAnalyst 5.0

Name: guest5156195695692547889

October 4, 2021

1 Data Processing and Normalization

1.1 Reading and Processing the Raw Data

MetaboAnalyst accepts a variety of data types generated in metabolomic studies, including compound concentration data, binned NMR/MS spectra data, NMR/MS peak list data, as well as MS spectra (NetCDF, mzXML, mzDATA). Users need to specify the data types when uploading their data in order for MetaboAnalyst to select the correct algorithm to process them. Table 1 summarizes the result of the data processing steps.

1.1.1 Reading Concentration Data

The concentration data should be uploaded in comma separated values (.csv) format. Samples can be in rows or columns, with class labels immediately following the sample IDs.

Samples are in rows and features in columns The uploaded file is in comma separated values (.csv) format. The uploaded data file contains 72 (samples) by 24 (compounds) data matrix.

1.1.2 Data Integrity Check

Before data analysis, a data integrity check is performed to make sure that all the necessary information has been collected. The class labels must be present and contain only two classes. If samples are paired, the class label must be from -n/2 to -1 for one group, and 1 to n/2 for the other group (n is the sample number and must be an even number). Class labels with same absolute value are assumed to be pairs. Compound concentration or peak intensity values should all be non-negative numbers. By default, all missing values, zeros and negative values will be replaced by the half of the minimum positive value found within the data (see next section)

1.1.3 Missing value imputations

Too many zeroes or missing values will cause difficulties for downstream analysis. MetaboAnalyst offers several different methods for this purpose. The default method replaces all the missing and zero values with a small values (the half of the minimum positive values in the original data) assuming to be the detection limit. The assumption of this approach is that most missing values are caused by low abundance metabolites (i.e.below the detection limit). In addition, since zero values may cause problem for data normalization (i.e. log), they are also replaced with this small value. User can also specify other methods, such as replace by mean/median, or use K-Nearest Neighbours (KNN), Probabilistic PCA (PPCA), Bayesian PCA (BPCA) method, Singular Value Decomposition (SVD) method to impute the missing values [1]. Please choose the one that is the most appropriate for your data.

[1] Stacklies W, Redestig H, Scholz M, Walther D, Selbig J. *pcaMethods: a bioconductor package, providing PCA methods for incomplete data.*, Bioinformatics 2007 23(9):1164-1167

Zero or missing values were replaced by 1/5 of the min positive value for each variable.

1.1.4 Data Filtering

The purpose of the data filtering is to identify and remove variables that are unlikely to be of use when modeling the data. No phenotype information are used in the filtering process, so the result can be used with any downstream analysis. This step can usually improves the results. Data filter is strongly recommended for datasets with large number of variables (> 250) datasets contain much noise (i.e.chemometrics data). Filtering can usually improve your results[2].

For data with number of variables < 250, this step will reduce 5% of variables; For variable number between 250 and 500, 10% of variables will be removed; For variable number bwteen 500 and 1000, 25% of variables will be removed; And 40% of variabled will be removed for data with over 1000 variables. The None option is only for less than 5000 features. Over that, if you choose None, the IQR filter will still be applied. In addition, the maximum allowed number of variables is ***10000***

No data filtering was performed.

[2]Hackstadt AJ, Hess AM.*Filtering for increased power for microarray data analysis*, BMC Bioinformatics. 2009; 10: 11.

Table 1: Summary of data processing results

	Features (positive)	Missing/Zero	Features (processed)
AM inf 24_1	22	2	24
AM inf 24_2	22	2	24
AM inf 24_3	23	1	24
AM inf 24_4	24	0	24
AM inf 24_5	22	2	24
AM inf 24_6	20	4	24
AM inf 48_1	23	1	24
AM inf 48_2	22	2	24
AM inf 48_3	21	3	24
AM inf 48_4	22	2	24
AM inf 48_5	21	3	24
AM inf 48_6	22	2	24
AM inf 6_1	21	3	24
AM inf 6_2	22	2	24
AM inf 6_3	21	3	24
AM inf 6_4	21	3	24
AM inf 6_5	23	1	24
AM inf 6_6	23	1	24
AM 24_1	21	3	24
AM 24_2	22	2	24
AM 24_3	24	0	24
AM 24_4	23	1	24
AM 24_5	22	2	24
AM 24_6	23	1	24
AM 48_1	22	2	24
AM 48_2	22	2	24
AM 48_3	23	1	24
AM 48_4	22	2	24
AM 48_5	23	1	24
AM 48_6	22	2	24
AM 6_1	20	4	24
AM 6_2	20	4	24
AM 6_3	20	4	24
AM 6_4	21	3	24
AM 6_5	23	1	24
AM 6_6	24	0	24
W inf 24_1	21	3	24
W inf 24_2	21	3	24
W inf 24_3	24	0	24
W inf 24_4	22	2	24
W inf 24_5	19	5	24
W inf 24_6	22	2	24
W inf 48_1	23	1	24
W inf 48_2	21	3	24
W inf 48_3	23	1	24
W inf 48_4	21	3	24
W inf 48_5	22	2	24
W inf 48_6	22	2	24
W inf 6_1	22	2	24
W inf 6_2	22	2	24
W inf 6_3	21	3	24
W inf 6_4	20	4	24
W inf 6_5	23	1	24
W inf 6_6	23	1	24
Water 24_1	21	3	24
Water 24_2	20	4	24
Water 24_3	24	0	24
Water 24_4	23	1	24
Water 24_5	23	1	24
Water 24_6	21	3	24
Water 48_1	20	4	24
Water 48_2	21	3	24
Water 48_3	22	2	24
Water 48_4	24	0	24
Water 48_5	21	3	24
Water 48_6	21	3	24
Water 6_1	22	2	24
Water 6_2	23	1	24
Water 6_3	22	2	24
Water 6_4	21	3	24
Water 6_5	22	2	24
Water 6_6	24	0	24

1.2 Data Normalization

The data is stored as a table with one sample per row and one variable (bin/peak/metabolite) per column. The normalization procedures implemented below are grouped into four categories. Sample specific normalization allows users to manually adjust concentrations based on biological inputs (i.e. volume, mass); row-wise normalization allows general-purpose adjustment for differences among samples; data transformation and scaling are two different approaches to make features more comparable. You can use one or combine both to achieve better results.

The normalization consists of the following options:

1. Row-wise procedures:
    - Sample specific normalization (i.e. normalize by dry weight, volume)
    - Normalization by the sum
    - Normalization by the sample median
    - Normalization by a reference sample (probabilistic quotient normalization)[3]
    - Normalization by a pooled or average sample from a particular group
    - Normalization by a reference feature (i.e. creatinine, internal control)
    - Quantile normalization
2. Data transformation :
    - Generalized log transformation (glog 2)
    - Cube root transformation
3. Data scaling:
    - Mean centering (mean-centered only)
    - Auto scaling (mean-centered and divided by standard deviation of each variable)
    - Pareto scaling (mean-centered and divided by the square root of standard deviation of each variable)
    - Range scaling (mean-centered and divided by the value range of each variable)

Figure 1 shows the effects before and after normalization.

[3]Dieterle F, Ross A, Schlotterbeck G, Senn H. *Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. Application in 1H NMR metabonomics*, 2006, Anal Chem 78 (13);4281 - 4290

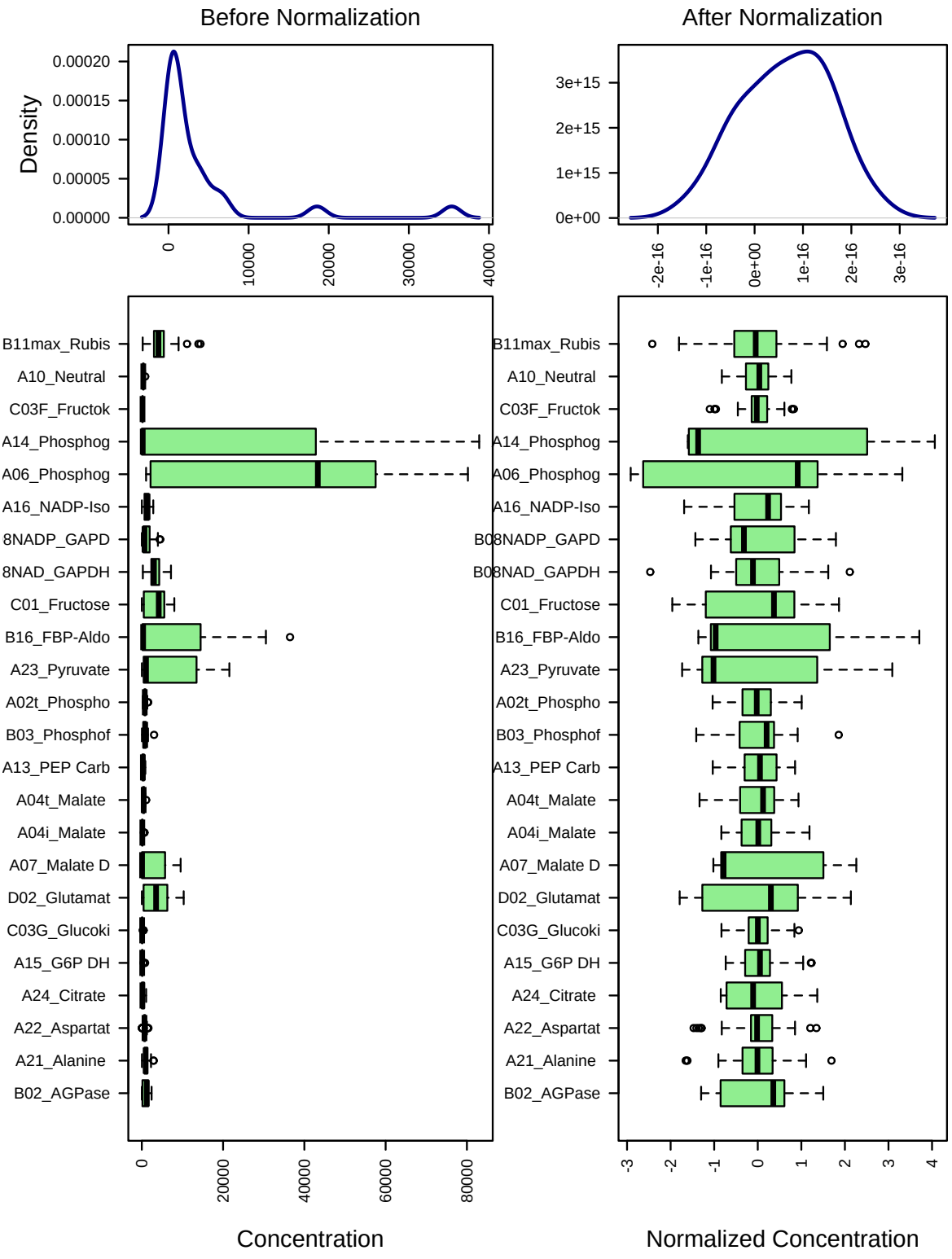

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples. Selected methods : Row-wise normalization: Normalization to sample median; Data transformation: Square Root Transformation; Data scaling: Pareto Scaling.

2 Statistical and Machine Learning Data Analysis

MetaboAnalyst offers a variety of methods commonly used in metabolomic data analyses. They include:

1. Univariate analysis methods:
    - Fold Change Analysis
    - T-tests
    - Volcano Plot
    - One-way ANOVA and post-hoc analysis
    - Correlation analysis
2. Multivariate analysis methods:
    - Principal Component Analysis (PCA)
    - Partial Least Squares - Discriminant Analysis (PLS-DA)
3. Robust Feature Selection Methods in microarray studies
    - Significance Analysis of Microarray (SAM)
    - Empirical Bayesian Analysis of Microarray (EBAM)
4. Clustering Analysis
    - Hierarchical Clustering
        - Dendrogram
        - Heatmap
    - Partitional Clustering
        - K-means Clustering
        - Self-Organizing Map (SOM)
5. Supervised Classification and Feature Selection methods
    - Random Forest
    - Support Vector Machine (SVM)

`Please note: some advanced methods are available only for two-group sample analyais.`

2.1 One-way ANOVA

Univariate analysis methods are the most common methods used for exploratory data analysis. For multi-group analysis, MetaboAnalyst provides one-way Analysis of Variance (ANOVA). As ANOVA only tells whether the overall comparison is significant or not, it is usually followed by post-hoc analyses in order to identify which two levels are different. MetaboAnalyst provides two most commonly used methods for this purpose - Fisher's least significant difference method (Fisher's LSD) and Tukey's Honestly Significant Difference (Tukey's HSD). The univariate analyses provide a preliminary overview about features that are potentially significant in discriminating the conditions under study.

Figure 2 shows the important features identified by ANOVA analysis. Table 2 shows the details of these features. The `post-hoc Sig. Comparison` column shows the comparisons between different levels that are significant given the p value threshold.

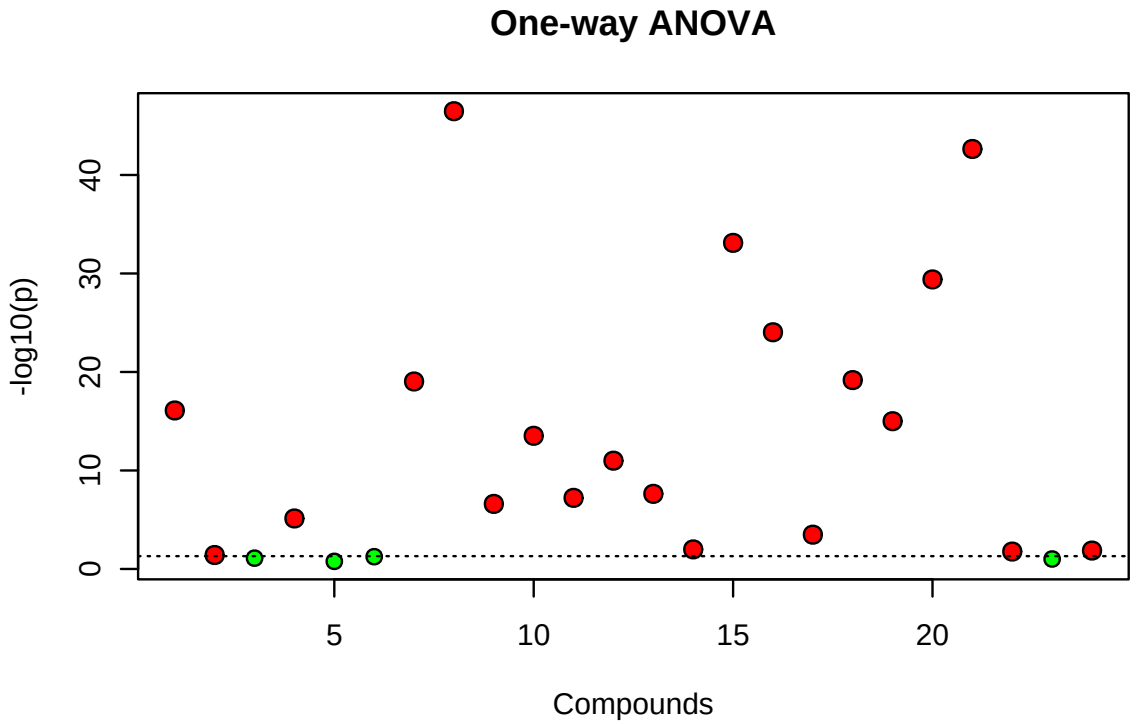

Figure 2: Important features selected by ANOVA plot with p value threshold 0.05.

Table 2: Important features identified by One-way ANOVA and post-hoc analysis

	Compounds	f.value	p.value	-log10(p)	FDR	Fisher's LSD
1	A07_Malate DH (NAD)	279.42	0.00	46.48	0.00	AM inf6 - AM inf24; AM6 - AM inf24; W inf6 - AM inf24;
2	A14_Phosphoglycerokinase_direct	206.56	0.00	42.64	0.00	AM inf6 - AM inf24; AM6 - AM inf24; W inf6 - AM inf24;
3	B16_FBP-Aldolase	96.13	0.00	33.11	0.00	AM inf6 - AM inf24; AM6 - AM inf24; W inf6 - AM inf24;
4	A06_Phosphoglucomutase	70.73	0.00	29.40	0.00	AM inf24 - AM inf6; AM inf24 - AM6; AM inf24 - W inf6;
5	C01_Fructose-1,6-bisphosphatase (cyt)	44.73	0.00	24.03	0.00	AM inf24 - AM inf6; AM inf24 - AM6; AM inf24 - W inf6;
6	B08NADP_GAPDH (NADP)	28.82	0.00	19.18	0.00	AM inf6 - AM inf24; AM6 - AM inf24; W inf6 - AM inf24;
7	D02_Glutamate DH (NAD)	28.44	0.00	19.04	0.00	AM inf24 - AM inf6; AM inf24 - AM6; AM inf24 - W inf6;
8	B02_AGPase	21.39	0.00	16.09	0.00	AM inf24 - AM inf48; AM inf24 - AM inf6; AM inf24 - AM
9	A16_NADP-Isocitrate DH	19.15	0.00	15.00	0.00	AM inf24 - AM inf6; AM inf24 - AM6; AM inf24 - W inf6;
10	A04t_Malate DH total (NADP)	16.39	0.00	13.52	0.00	AM inf24 - AM inf6; AM48 - AM inf24; AM inf24 - AM6; A
11	B03_Phosphofructokinase (PPi)	12.34	0.00	10.99	0.00	AM inf24 - AM inf6; AM inf24 - AM6; AM inf24 - W inf6;
12	A02t_Phosphoglucose Isomerase (total)	7.99	0.00	7.63	0.00	AM inf6 - AM inf24; AM48 - AM inf24; AM6 - AM inf24; W
13	A13_PEP Carboxylase	7.53	0.00	7.22	0.00	AM inf6 - AM inf24; AM6 - AM inf24; W inf6 - AM inf24;
14	A04i_Malate DH initial (NADP)	6.86	0.00	6.61	0.00	AM inf24 - AM inf6; AM inf24 - AM6; AM inf24 - W inf6;
15	A24_Citrate synthase	5.36	0.00	5.13	0.00	AM inf48 - AM inf24; AM inf6 - AM inf24; AM6 - AM inf2
16	B08NAD_GAPDH (NAD)	3.84	0.00	3.48	0.00	AM inf24 - W inf48; Water6 - AM inf24; AM inf48 - W inf
17	A23_Pyruvate kinase	2.55	0.01	1.99	0.01	AM inf24 - AM inf6; AM inf24 - AM6; AM inf24 - W inf6;
18	B11max_Rubisco (maximal)	2.45	0.01	1.87	0.02	AM inf24 - W inf48; AM6 - AM inf48; AM inf48 - W inf48;
19	C03F_Fructokinase	2.36	0.02	1.77	0.02	AM inf24 - W inf24; AM inf24 - W inf48; AM inf6 - AM in
20	A21_Alanine aminotransferase	2.05	0.04	1.42	0.05	AM inf24 - AM48; AM inf6 - AM inf48; W inf6 - AM inf48

2.2 Correlation Analysis

Correlation analysis can be used to visualize the overall correlations between different features It can also be used to identify which features are correlated with a feature of interest. Correlation analysis can also be used to identify if certain features show particular patterns under different conditions. Users first need to define a pattern in the form of a series of hyphenated numbers. For example, in a time-series study with four time points, a pattern of of `1-2-3-4` is used to search compounds with increasing the concentration as time changes; while a pattern of `3-2-1-3` can be used to search compounds that decrease at first, then bounce back to the original level.

Figure 3 shows the overall correlation heatmap.

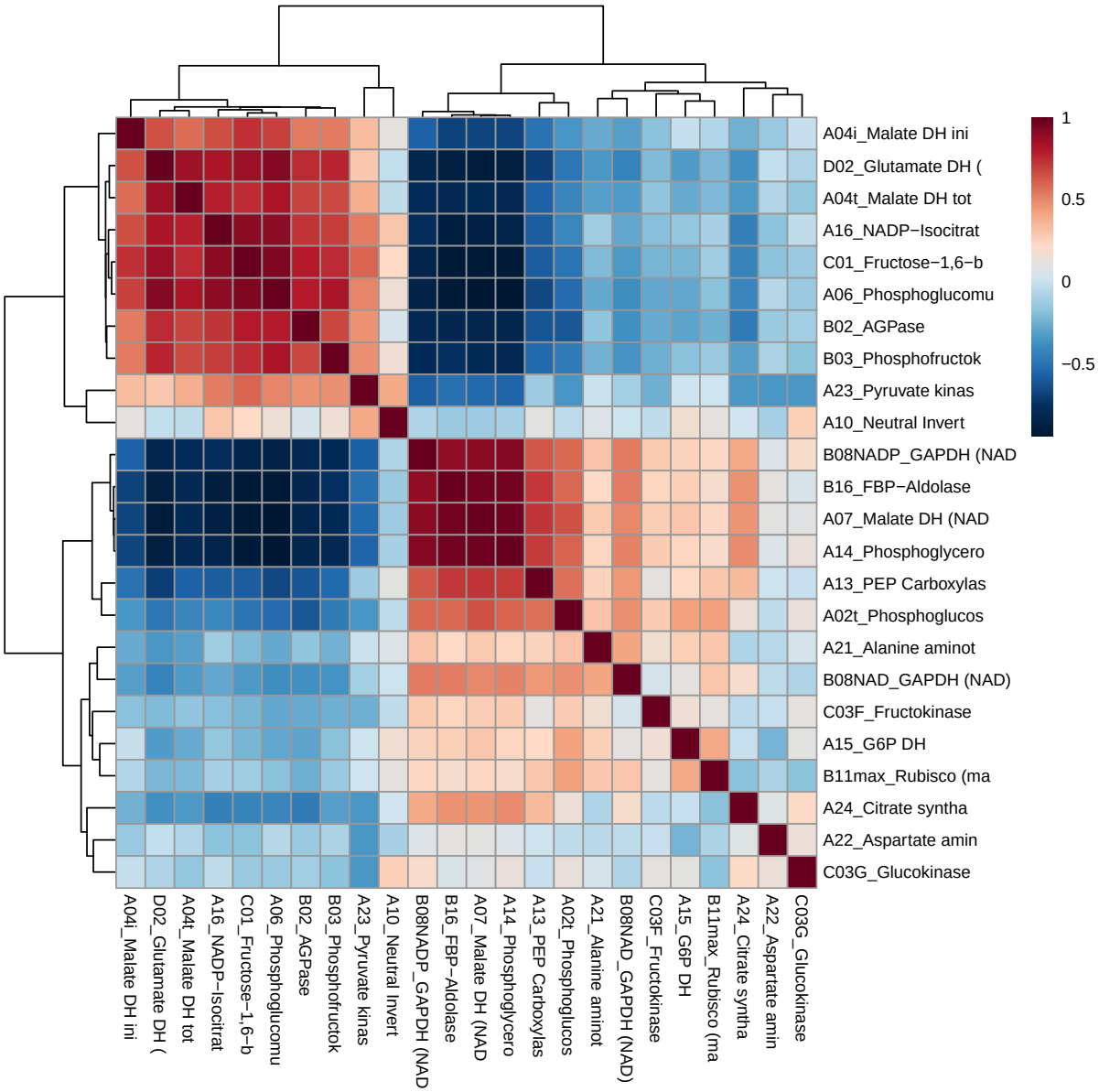

Figure 3: Correlation Heatmaps

2.3 Principal Component Analysis (PCA)

PCA is an unsupervised method aiming to find the directions that best explain the variance in a data set (X) without referring to class labels (Y). The data are summarized into much fewer variables called *scores* which are weighted average of the original variables. The weighting profiles are called *loadings*. The PCA analysis is performed using the `prcomp` package. The calculation is based on singular value decomposition.

The Rscript `chemometrics.R` is required. Figure 4 is pairwise score plots providing an overview of the various seperation patterns among the most significant PCs; Figure 5 is the scree plot showing the variances explained by the selected PCs; Figure 6 shows the 2-D scores plot between selected PCs; Figure 7 shows the 3-D scores plot between selected PCs; Figure 8 shows the loadings plot between the selected PCs; Figure 9 shows the biplot between the selected PCs.

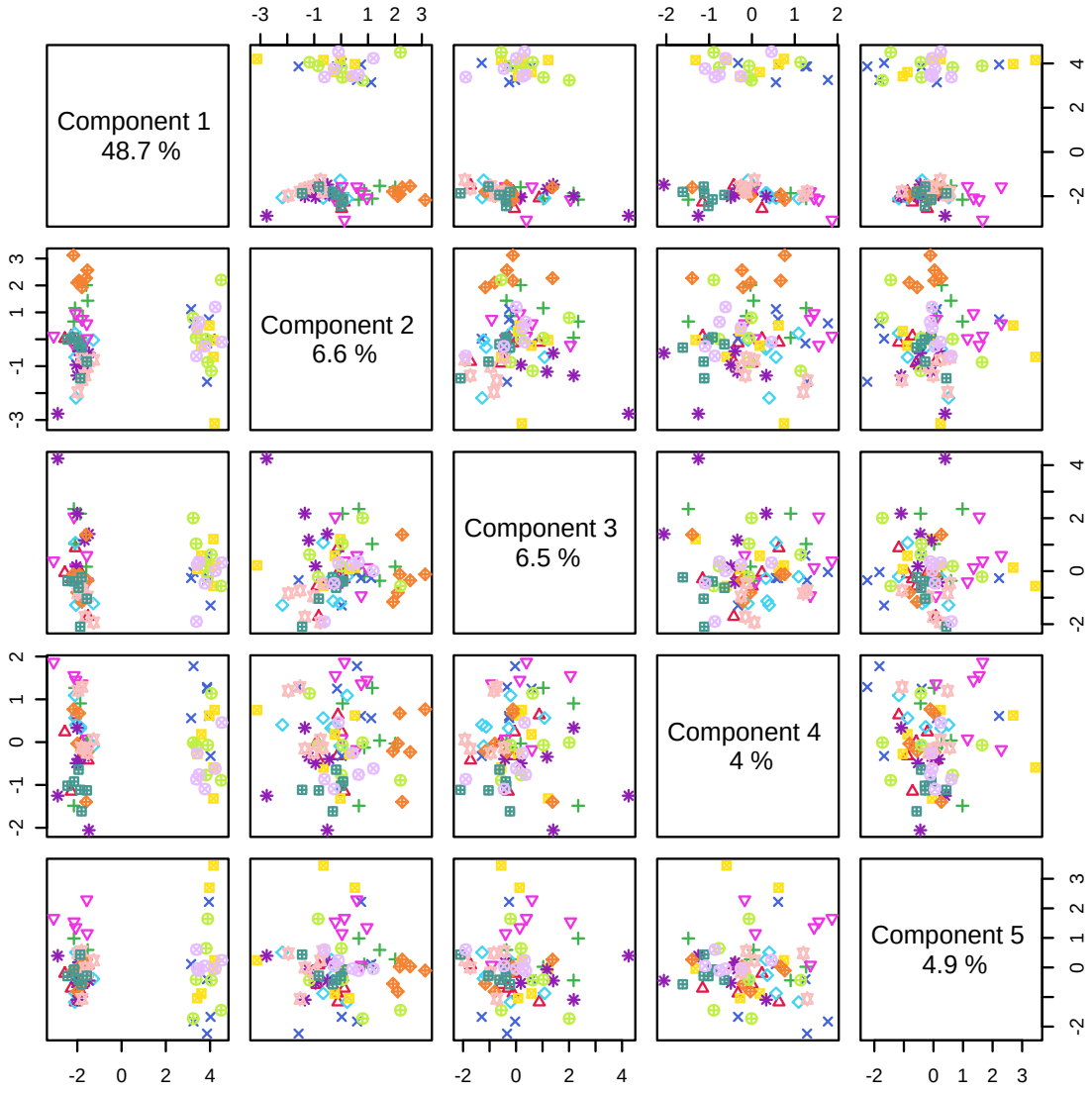

Figure 4: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

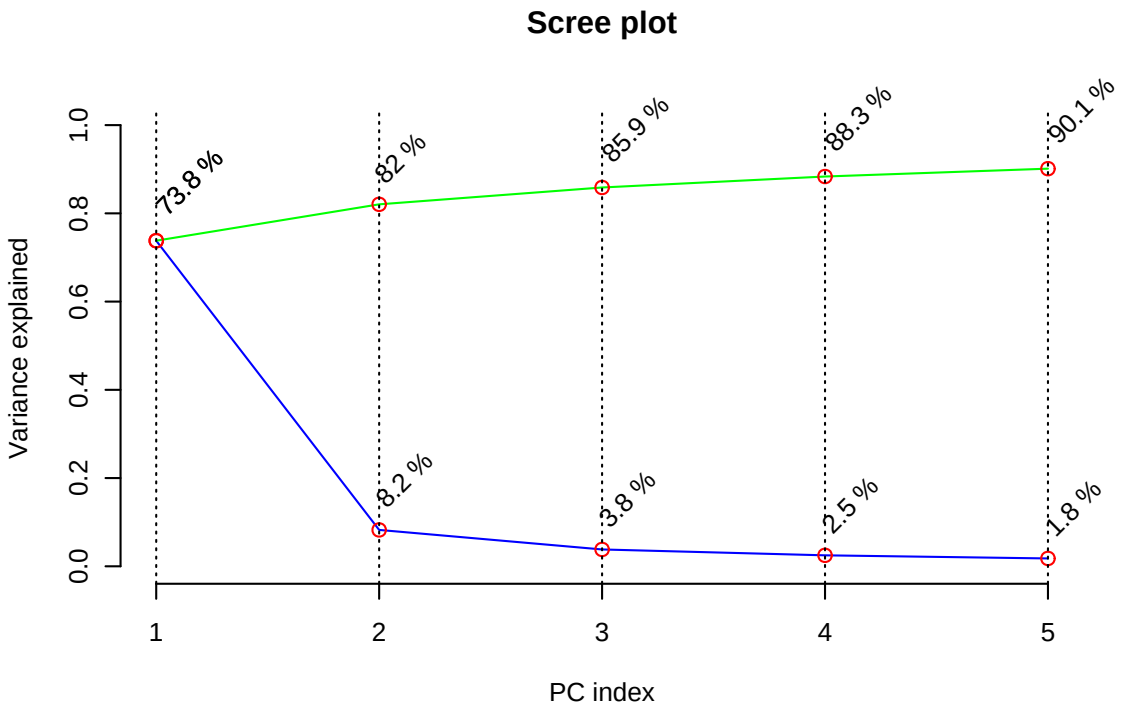

Figure 5: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

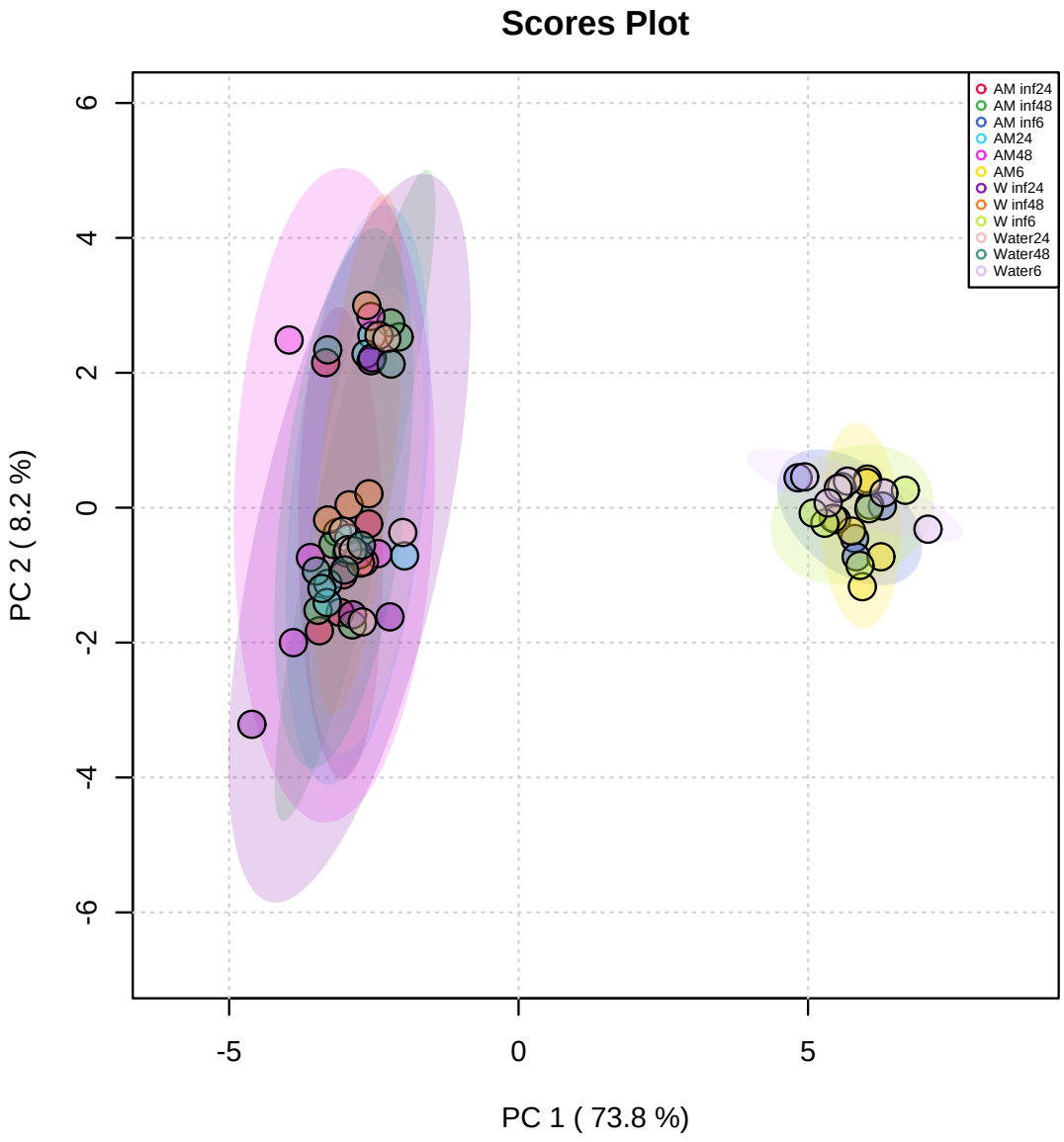

Figure 6: Scores plot between the selected PCs. The explained variances are shown in brackets.

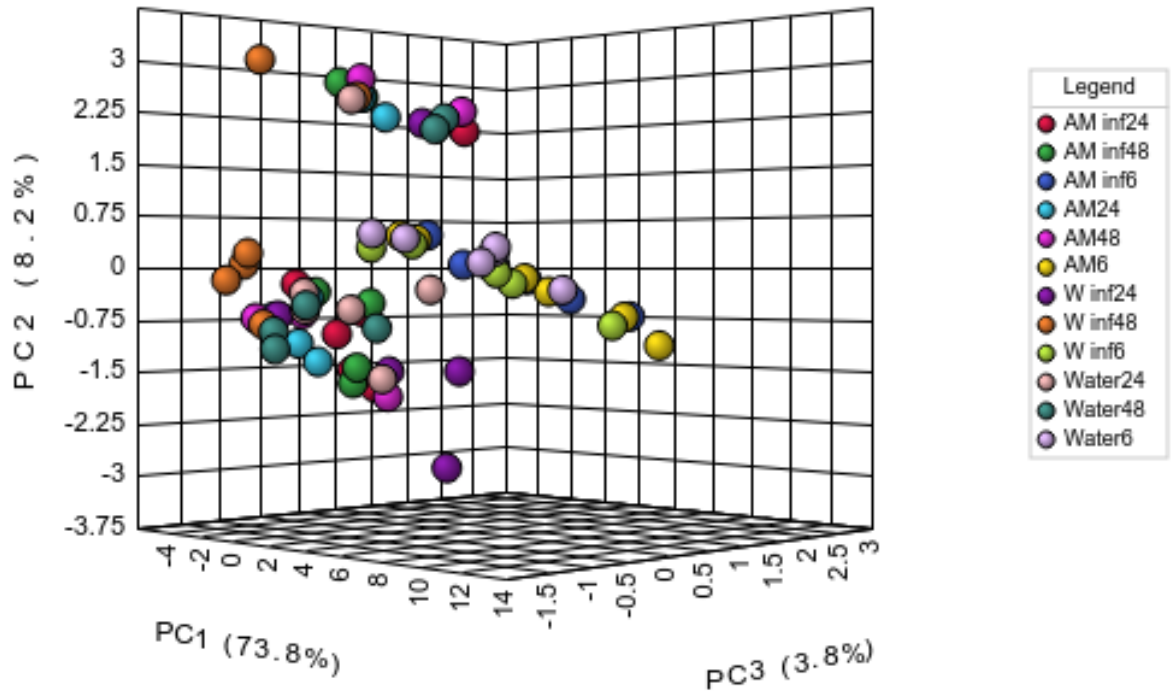

Figure 7: 3D score plot between the selected PCs. The explained variances are shown in brackets.

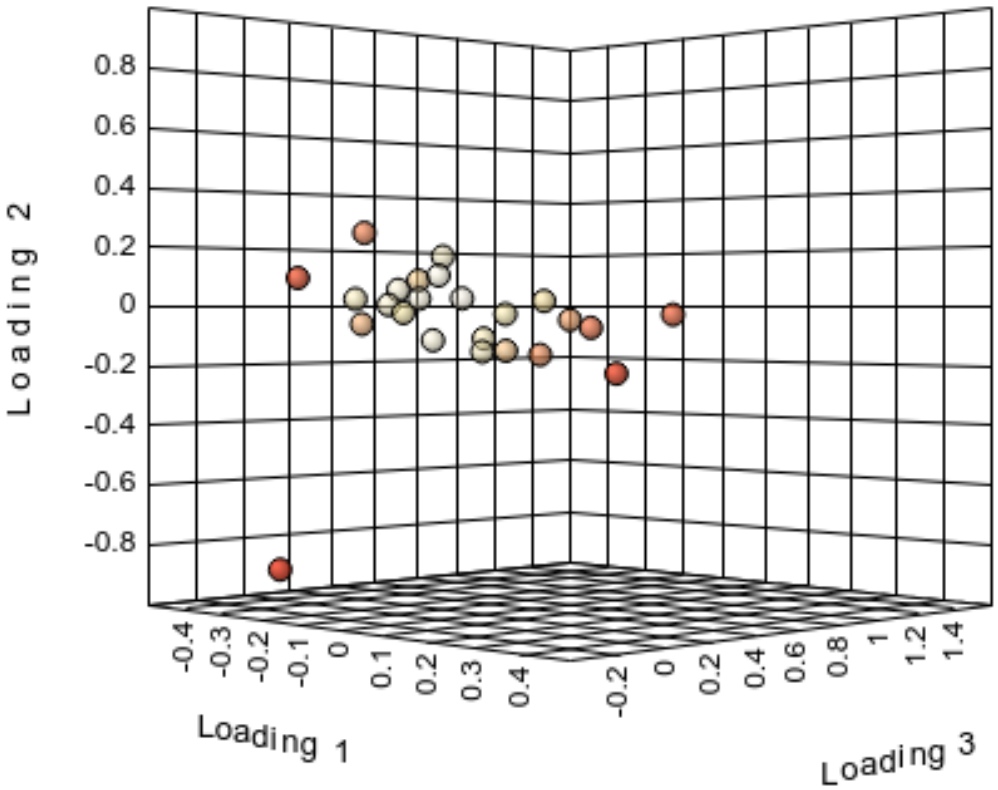

Figure 8: Loadings plot for the selected PCs.

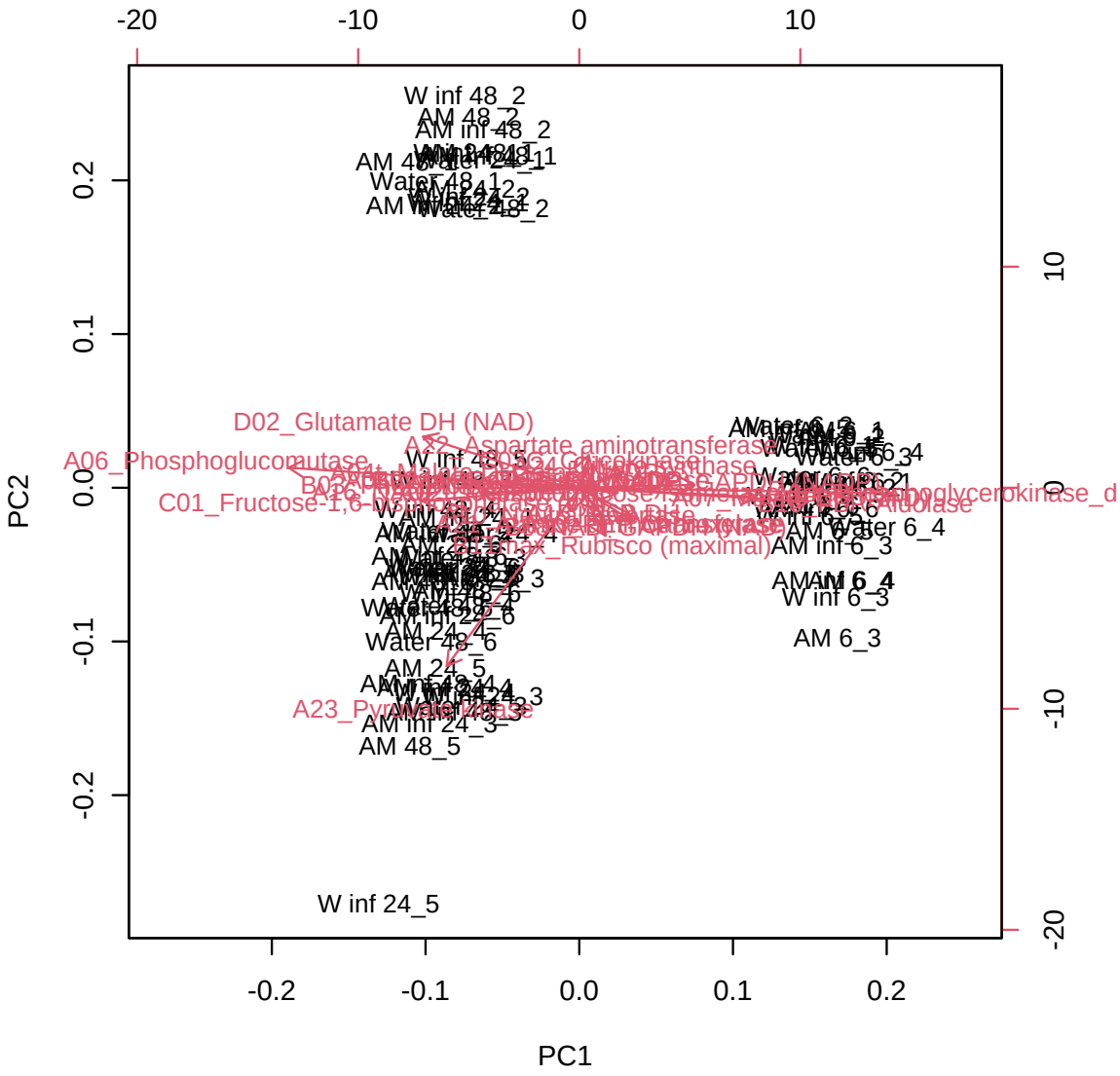

Figure 9: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

2.4 Sparse Partial Least Squares - Discriminant Analysis (sPLS-DA)

The sparse PLS-DA (sPLS-DA) algorithm can be used to effectively reduce the number of variables (metabolites) in high-dimensional metabolomics data to produce robust and easy-to-interpret models. Users can control the sparseness of the model by controlling the number of components in the model and the number of variables in each component. For more information, please refer to Cao et al. 2011 (PMC3133555).

Figure 10 shows the overview of scores plots; Figure 11 shows the 2-D scores plot between selected components; Figure 12 shows the loading plot of the top ranked features; Figure 13 shows the 3-D scores plot between selected components; Figure 14 shows the performance of the sPLS-DA model evaluated using cross-validations;

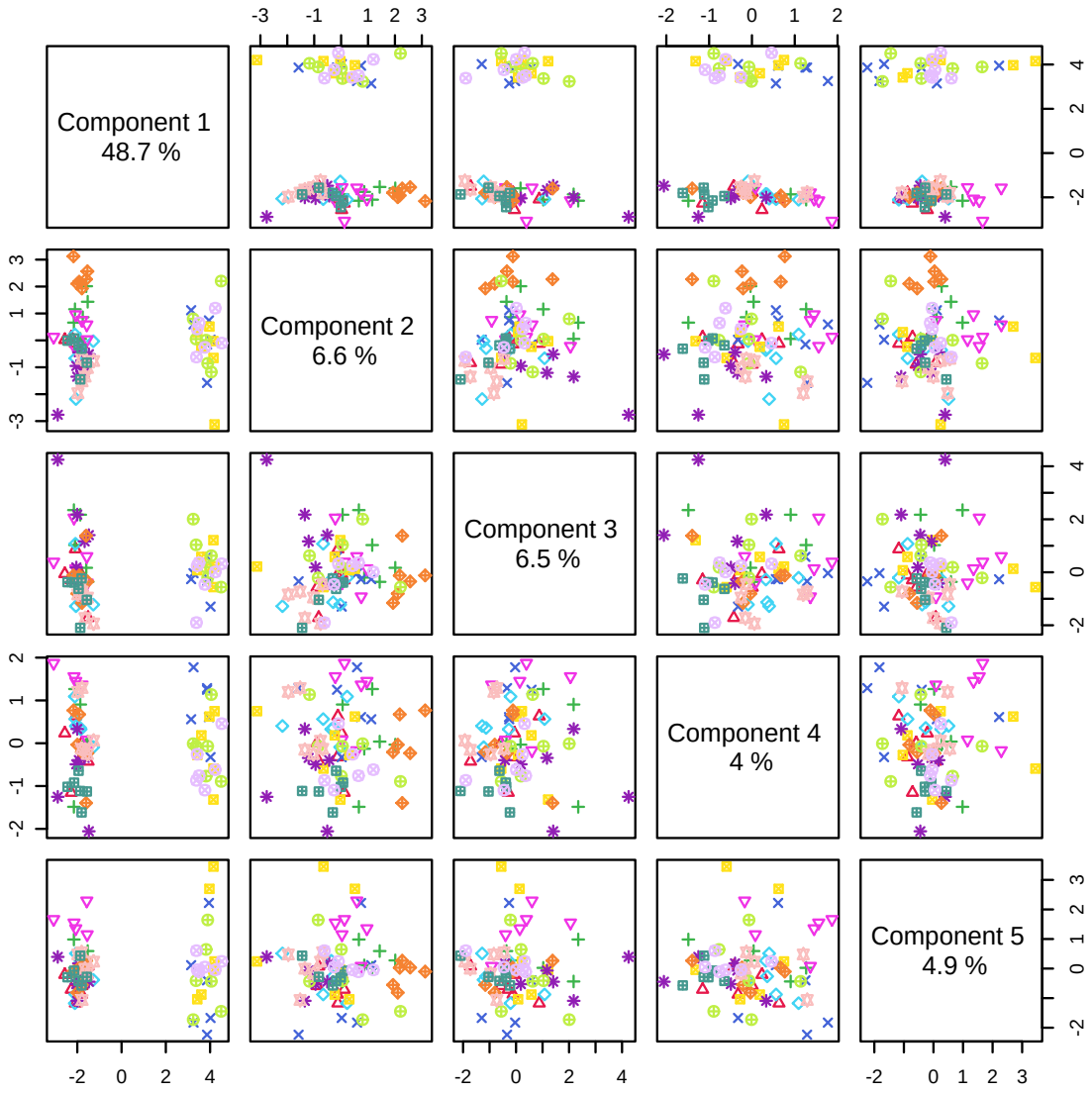

Figure 10: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

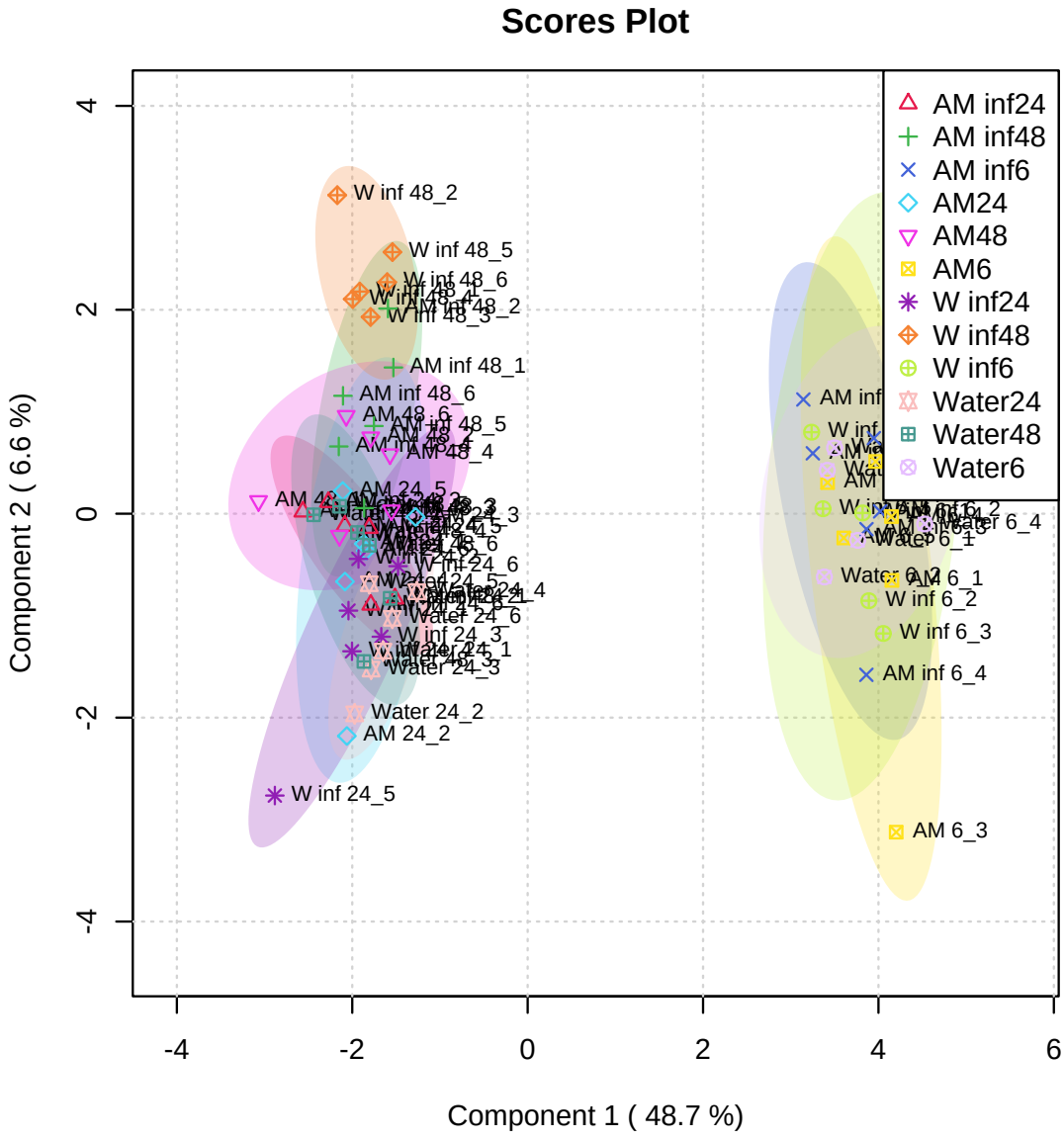

Figure 11: Scores plot between the selected PCs. The explained variances are shown in brackets.

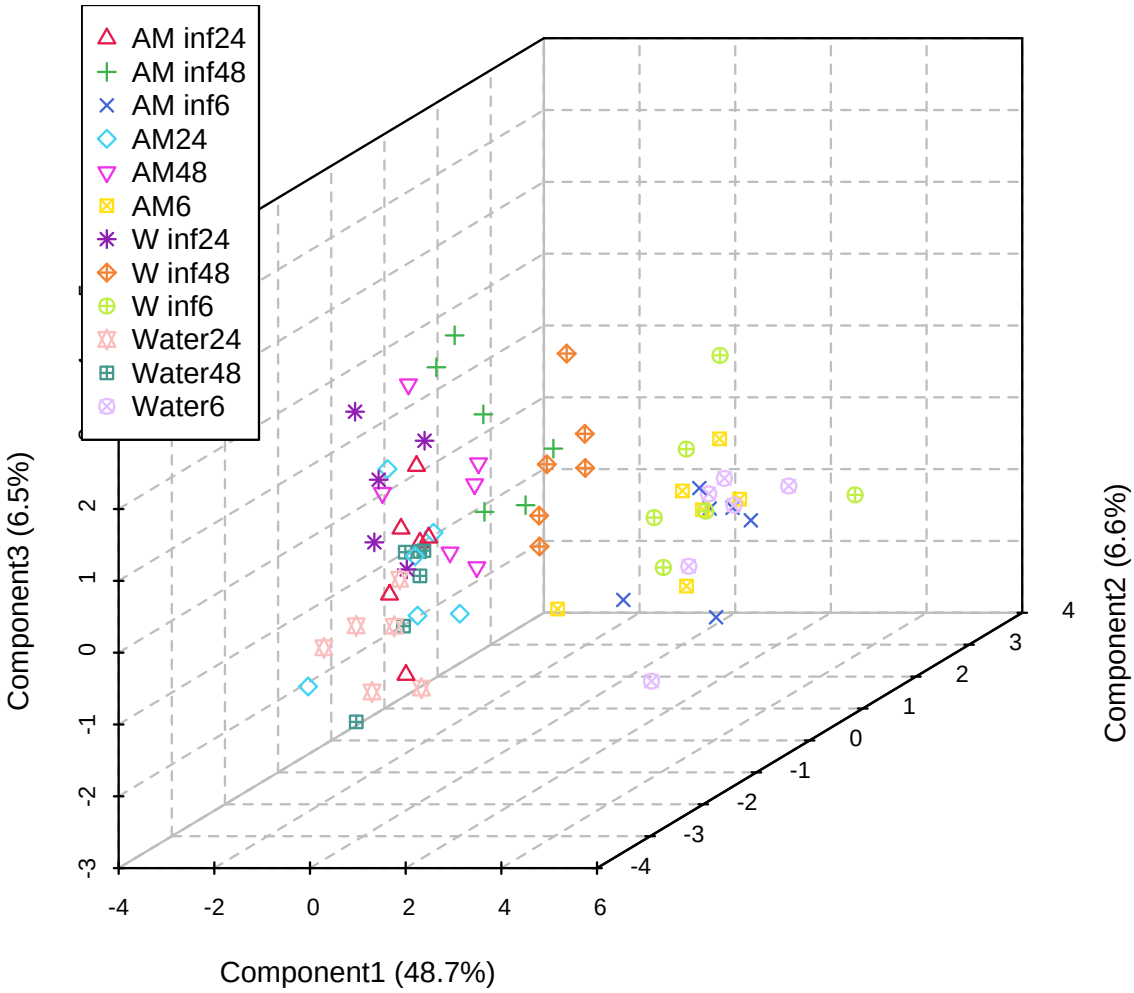

Figure 12: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

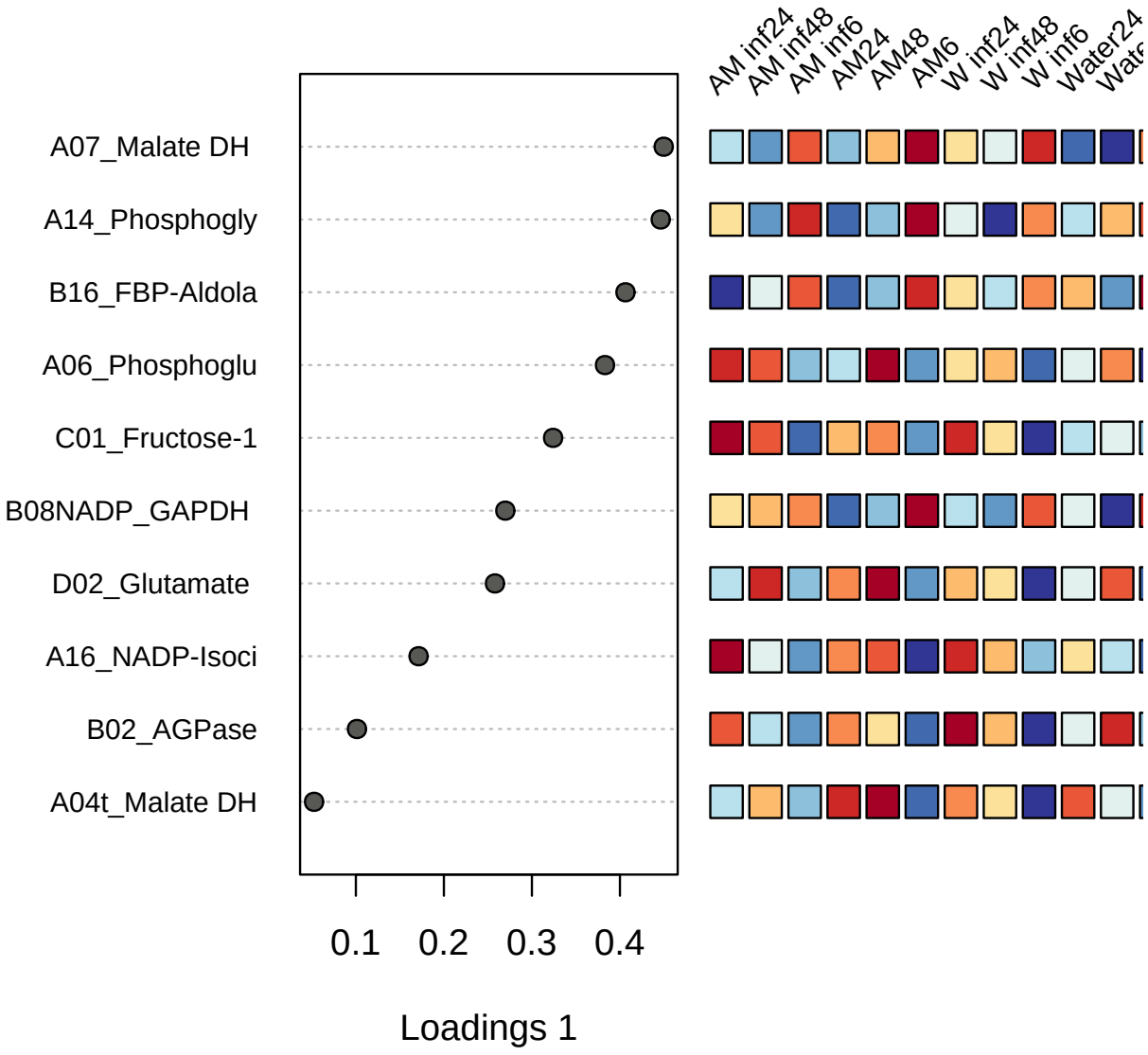

Figure 13: Plot showing the variables selected by the sPLS-DA model for a given component. The variables are ranked by the absolute values of their loadings.

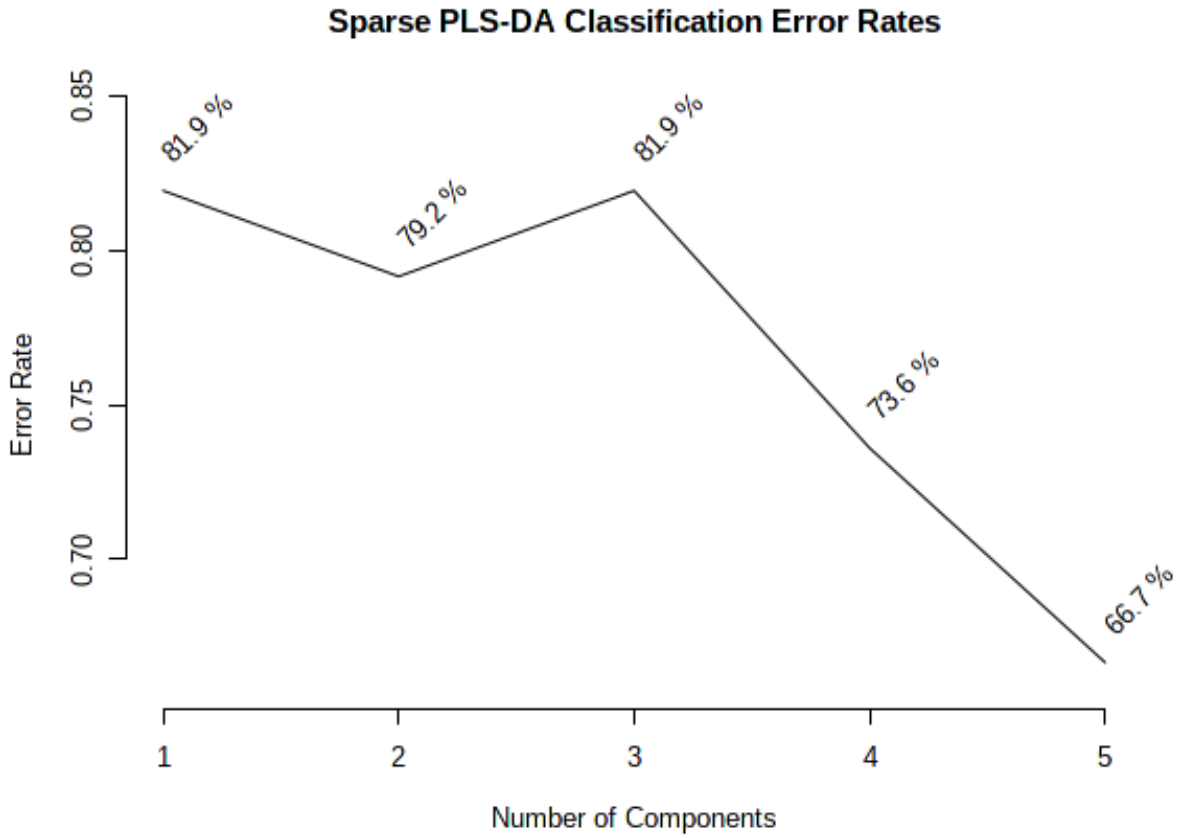

Figure 14: Plot of the performance of the sPLS-DA model evaluated using cross validations (CV) with increasing numbers of components created using the specified number of the variables. The error rate is on the y-axis and the number of components is on the x-axis.

2.5 Hierarchical Clustering

In (agglomerative) hierarchical cluster analysis, each sample begins as a separate cluster and the algorithm proceeds to combine them until all samples belong to one cluster. Two parameters need to be considered when performing hierarchical clustering. The first one is similarity measure - Euclidean distance, Pearson's correlation, Spearman's rank correlation. The other parameter is clustering algorithms, including average linkage (clustering uses the centroids of the observations), complete linkage (clustering uses the farthest pair of observations between the two groups), single linkage (clustering uses the closest pair of observations) and Ward's linkage (clustering to minimize the sum of squares of any two clusters). Heatmap is often presented as a visual aid in addition to the dendrogram.

Hierachical clustering is performed with the `hclust` function in package `stat`. Figure 15 shows the clustering result in the form of a dendrogram. Figure 16 shows the clustering result in the form of a heatmap.

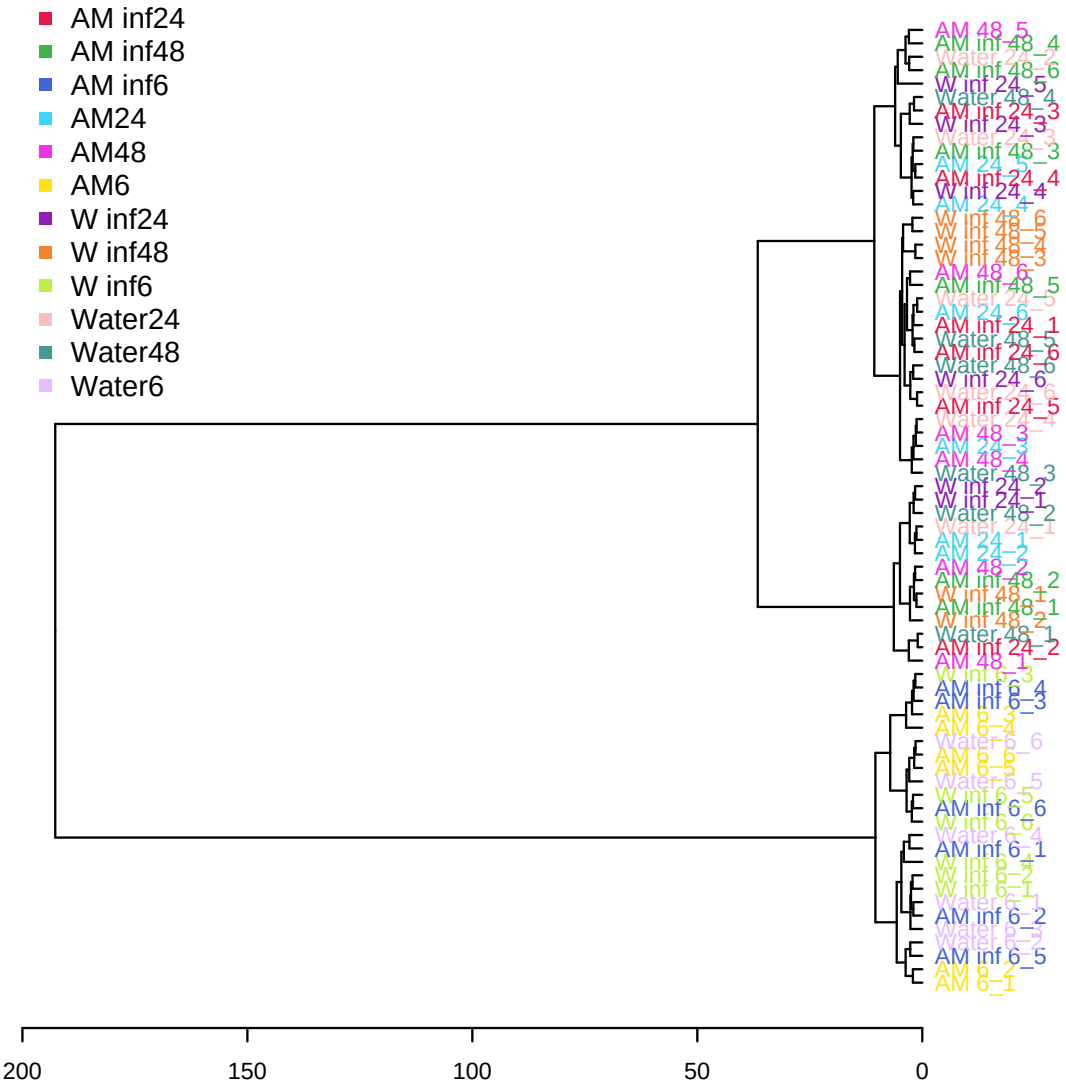

Figure 15: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

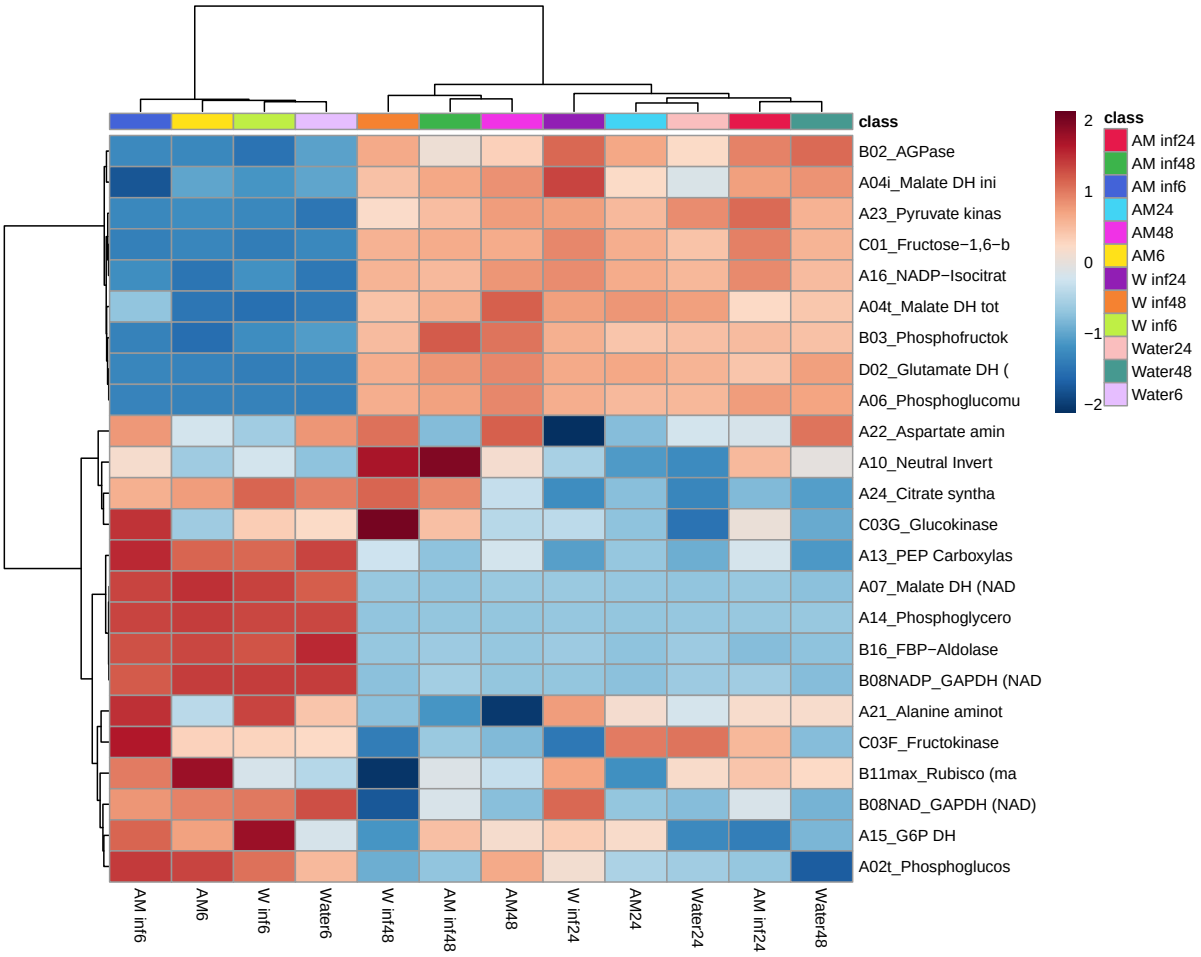

Figure 16: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

3 Appendix: R Command History

```
 [1] "mSet<-InitDataObjects(\"conc\", \"stat\", FALSE)"
 [2] "mSet<-Read.TextData(mSet, \"Replacing_with_your_file_path\", \"rowu\", \"disc\");"
 [3] "mSet<-SanityCheckData(mSet)"
 [4] "mSet<-ReplaceMin(mSet);"
 [5] "mSet<-PreparePrenormData(mSet)"
 [6] "mSet<-Normalization(mSet, \"MedianNorm\", \"SrNorm\", \"NULL\", ratio=FALSE, ratioNum=20)"
 [7] "mSet<-PlotNormSummary(mSet, \"norm_0_\", \"png\", 72, width=NA)"
 [8] "mSet<-PlotSampleNormSummary(mSet, \"snorm_0_\", \"png\", 72, width=NA)"
 [9] "mSet<-Normalization(mSet, \"MedianNorm\", \"SrNorm\", \"ParetoNorm\", ratio=FALSE, ratioNum=20)
[10] "mSet<-PlotNormSummary(mSet, \"norm_1_\", \"png\", 72, width=NA)"
[11] "mSet<-PlotSampleNormSummary(mSet, \"snorm_1_\", \"png\", 72, width=NA)"
[12] "mSet<-ANOVA.Anal(mSet, F, 0.05, \"fisher\", FALSE)"
[13] "mSet<-PlotANOVA(mSet, \"aov_0_\", \"png\", 72, width=NA)"
[14] "mSet<-UpdateLoadingCmpd(mSet, \"A07_Malate DH (mSetD)\")"
[15] "mSet<-PlotCmpdView(mSet, \"A07_Malate DH (mSetD)\", \"png\", 72, width=NA)"
[16] "mSet<-UpdateLoadingCmpd(mSet, \"A07_Malate DH (mSetD)\")"
[17] "mSet<-PlotCmpdView(mSet, \"A07_Malate DH (mSetD)\", \"png\", 72, width=NA)"
[18] "mSet<-PlotCorrHeatMap(mSet, \"corr_0_\", \"png\", 72, width=NA, \"col\", \"pearson\", \"bwm\",
[19] "mSet<-FeatureCorrelation(mSet, \"pearson\", \"B02_AGPase\")"
[20] "mSet<-PlotCorr(mSet, \"ptn_1_\", \"feature\", \"png\", 72, width=NA)"
[21] "mSet<-PCA.Anal(mSet)"
[22] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[23] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[24] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[25] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[26] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[27] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[28] "mSet<-SPLSR.Anal(mSet, 5, 10, \"same\", \"Mfold\")"
[29] "mSet<-PlotSPLSPairSummary(mSet, \"spls_pair_0_\", \"png\", 72, width=NA, 5)"
[30] "mSet<-PlotSPLS2DScore(mSet, \"spls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[31] "mSet<-PlotSPLS3DScoreImg(mSet, \"spls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[32] "mSet<-PlotSPLSLoading(mSet, \"spls_loading_0_\", \"png\", 72, width=NA, 1,\"overview\");"
[33] "mSet<-PlotSPLSDA.Classification(mSet, \"spls_cv_0_\", \"png\", 72, width=NA)"
[34] "mSet<-PlotSPLS3DLoading(mSet, \"spls_loading3d_0_\", \"json\", 1,2,3)"
[35] "mSet<-PlotSPLS2DScore(mSet, \"spls_score2d_1_\", \"png\", 72, width=NA, 1,2,0.95,1,0)"
[36] "mSet<-PlotSPLS2DScore(mSet, \"spls_score2d_2_\", \"png\", 72, width=NA, 1,3,0.95,1,0)"
[37] "mSet<-PlotSPLS2DScore(mSet, \"spls_score2d_3_\", \"png\", 72, width=NA, 2,3,0.95,1,0)"
[38] "mSet<-PlotSPLS2DScore(mSet, \"spls_score2d_4_\", \"png\", 72, width=NA, 1,2,0.95,1,0)"
[39] "mSet<-PlotHCTree(mSet, \"tree_0_\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[40] "mSet<-PlotHeatMap(mSet, \"heatmap_0_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[41] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[42] "mSet<-ComputeDSPC(mSet)"
[43] "mSet<-CreateGraph(mSet)"
[44] "mSet<-SaveTransformedData(mSet)"
[45] "mSet<-PreparePDFReport(mSet, \"guest5156195695692547889\")\n"
```

The report was generated on Mon Oct 4 05:34:27 2021 with R version 4.0.2 (2020-06-22).